A NOVEL SYSTEMATIC MAPPING APPROACH FOR HIGHLY EFFICIENT MULTIPLEXED FIR-FILTER ARCHITECTURES

Wolfgang Wilhelm and Tobias G. Noll

Chair on Electrical Engineering and Computer Systems,
University of Technology RWTH Aachen, Germany
email: wilhelm@eecs.rwth-aachen.de

ABSTRACT

A systematic mapping approach leading to efficient VLSI-architectures for FIR-filters with a wide range of system parameters is presented. This approach is subdivided into two steps. In the first step the folding technique is applied at bit-level. The free parameters of this technique are then fixed in the second step according to guidelines which are derived from design-strategies for efficient VLSI-architectures. For many applications this approach leads to a reduced hardware complexity in comparison with state-of-the-art techniques. In addition, regularity and scalability of the resulting architectures keep the design effort small. In order to demonstrate the efficiency and the flexibility of this approach a new class of efficient time-shared FIR-filters for adaptive equalizing and a new class of efficient matched filters for rapid code acquisition in spread spectrum receivers are presented.

1. INTRODUCTION

Programmable FIR digital filters are important building blocks in many systems for real time signal processing. Applications that are of broad interest cover high-speed filters for video signal processing, adaptive pulse shaping or signal equalization as well as low-speed extremely high-order matched filters for spread spectrum systems. Thus, the generic throughput rate f_0 obtained for a certain circuit concept in a given technology by proper optimization of the efficiency seldom fits the required throughput rate f_s exactly. Hence, strategies which trade throughput rate for silicon area efficiently are highly desirable.

The requirements in high performance applications can be met by well known architectural concepts of parallelization (e.g. [1]). However, with the permanent progress of CMOS technologies architectural concepts of time-sharing become more and more important not only in high volume applications, but also in order to enable single chip solutions. Assuming a state-of-the-art 0.5-μm CMOS technology a transversal filter efficiently implemented can reach a generic throughput rate of up to 200 MHz. For Wireless Local Area Networks (WLAN), for example, utilizing the 902-928MHz ISM frequency band efficiently with a chip rate of 12.7 MChips/s [2] the matched filter could be used several times by a time-sharing factor of 16 so as to reduce the amount of chip area.

Typically the attempt to match low throughput requirements is performed by time space mapping [3, 4] using simple algebraic techniques. But in the general case of irregular [5] signal flow graphs (SFG) like Bit-Plane structures the mapping leads to nonlinear equations. Another drawback of this technique is the lack of a direct relation between design constraints and placement of processing elements in the SFG before mapping.

In contrast to that the folding technique applied at bit-level enables a systematic synthesis of efficient SFGs because demands on circuit or layout level can be directly mapped to the parameters of the algorithm.

2. BASIC STEPS OF THE DESIGN APPROACH

2.1. Folding Algorithm at Bit-Level

Although especially developed for DSP applications at word-level the folding technique is as universal as to be applicable at bit-level, too.

Following the notation introduced in [5] the terms algorithm SFG (aSFG) for the initial SFG and hardware SFG (hSFG) for the resulting SFG will be used (see Fig. 1a).
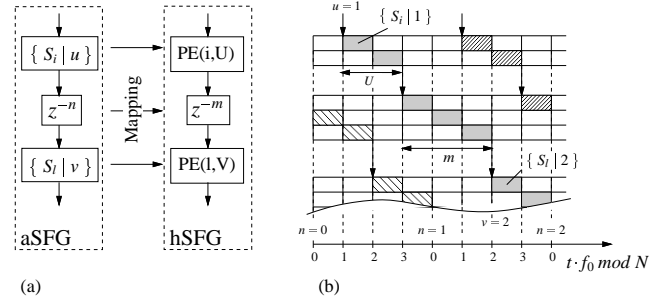

Fig. 1: (a) Definitions (b) Transformation of Edge Delays
$(N = f_0/f_s = 4,\ n = 1,\ u = 1,\ v = 2,\ U = 2)$

The first step of the technique is to separate the aSFG into so-called folding elements which contain one or more adjacent operations (e.g. adders, registers, partial products). For a required time-sharing factor of N, up to N equally structured folding elements are collected in a respective ordered so-called folding set. Each folding element belonging to the same folding set will be assigned to the same processing element (PE) in the hSFG. Within the execution sequence (folding order) of the folding elements on the assigned PE a folding order number is defined for each folding element. This folding order number is defined to be the time partition in which the task of the folding element is executed modulo the iteration period N. In Fig. 1a the notation $\{S_i|u\}$ is used for a folding element belonging to the set S_i with folding order number u.

After determining an appropriate level of pipelining for each PE in the hSFG the second step of the folding technique consists of transforming the delays in the edges between operations in the aSFG to the hSFG. As demonstrated by the sample scheduling scheme in Fig. 1b, n delays will be transformed in $m = n \cdot N - U + v - u$ delays. In order to ensure causality in the third step of the technique all transformed edge delays have to be made non-negative by means of cut-set transformations where necessary.

2.2. Guidelines to Achieve High Efficiency

Our design strategy consists of a consequent spatial separation of optimally pipelined functional blocks (datapaths) and blocks for synchronizing data (synchronizing networks). Separate optimization of datapaths and synchronizing networks will then lead to high clock frequencies, small amounts of chip area, and low power consumption as a result of short local interconnections. Moreover, synchronizing networks can be shared for multiple datapath units.

Since datapaths determine the maximum clock frequency and usually occupy the largest part of the overall silicon area the supposed design criteria are focussed on the optimization of the datapaths. These criteria and their consideration in appropriate folding sets and folding orders will be discussed briefly.

The **first** criterion concerns the cost of multiplexers and feedback loops which are inevitably introduced by the folding technique. In order to limit the number of multiplexers at the first stage of the datapath and the number of feedback paths (see e.g. Fig. 2b) to one,

- spatially adjacent folding elements in the aSFG must belong to different folding sets (except only one folding set is used),

- looking at successive folding elements in the aSFG the associated folding sets must build a recurrent sequence,

- the difference of folding order numbers between any folding element (except the last) and its successor in the aSFG must be fixed for each folding set.

Due to the application of the folding technique at bit-level the PEs only contain adders and registers. Especially if the area compact 24-transistor full-adders in combination with transmission-gate latches are used, splitting each pipeline register into two latches distributed among two adder stages leads to particularly efficient architectures [6]. In order not to interrupt the structure of the datapath by additional synchronization delays the **second** criterion requires edges between two PEs without delays and PEs containing even numbers of adders.

As a **third** criterion the feedback path must contain at least one register so as not to become part of the critical path.

Let i_n be the count of delays in the adder chain between adjacent folding elements mapped to the n-th and $(n+1)$-th PE and k_n the number of adders in the first of these two folding elements. Then the second and third criterion require folding order number differences of

- $k_n/2 - i_nN$ between any folding element except those assigned to the last PE and its succeeding element,

- $k_n/2 + 1 - i_nN$ between any folding element assigned to the last PE and its succeeding element.

For filters with coefficient wordlength $w_c > 1$bit the **fourth** criterion is to minimize the internal adder wordlength without causing any accumulation errors. The consequences on algorithmic level are

- after stripping the coefficients into digits (with a fixed number of bits) partial products corresponding to the same digit have to be assigned to a common folding element,

- within any folding set the sequence of digits sorted by the associated folding order numbers must show increasing digit weights (except for the most significant digit which is followed by the least significant digit),

- within any folding element the sequence of coefficient bits has to be sorted by their weights

- a folding element with increasing coefficient weights has to be followed by a folding element with decreasing coefficient weights and vice versa

- in a folding element assigned to the n-th PE and its succeeding element, digits with the same weight have to be processed with an offset of $(k_n/2)$ mod N time partitions.

The **fifth** criterion concerns the cost of the routing for shifting the output signal of the last PE before feeding back. In order to minimize this cost, the weight of the first coefficient in a folding element assigned to the first PE have to differ from the weight of the last coefficient in an adjacent folding element assigned to the last PE by a factor of no more than two.

3. NOVEL CLASSES OF FIR-FILTER ARCHITECTURES

3.1. Adaptive Equalizer

Starting from a parallel-in/serial-out SFG at bit-level an appropriate rearrangement of coefficient weights leads to aSFGs that makes savings in silicon area for applications like adaptive equalizers possible. The example in Fig. 2 allows for the time-sharing factor of $N = 2$ to halve the number of adder stages as well as to reduce the internal adder wordlength by 4bit.
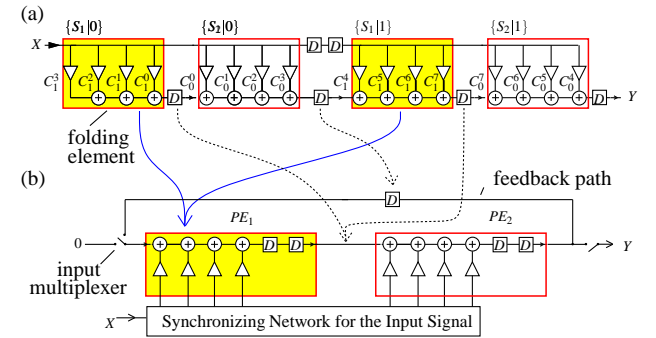

Fig. 2: (a) aSFG (b) hSFG ($M = 2$taps, $w_c = 8$bit, $N = 2$)

For most of the adder stages in the hSFG simple NAND-gates are sufficient in order to generate the partial products. Only for the last adder stage in each PE the signs of the coefficients have to be taken into account by additional XOR-gates. The multiplexers and XOR-gates of the architecture can be controlled by the same signal.

Applying the cut-set technique, the pipeline registers within each PE have to be distributed among the adder stages. The resulting pipelining for the incoming data can be realized with one

additional latch within each basic cell independent of the system parameters. Furthermore in addition to the feedback path only one wire has to be routed beside each basic cell.

The synchronizing network for the input signal contains at most one chain of registers clocked at the sampling rate, and a multiplexer. Since typically the adaption rate of equalizers is much smaller than the sample rate, the synchronizing network for the coefficients can be realized cost efficiently using a pointer-chain build up from a 1-bit shift register.

The use of carry-save arithmetic and transmission-gate latches in combination with a complementary two-phase clocking scheme results in a silicon area of $0.46mm^2$ for the 3taps FIR-filter shown in Fig. 3. The layout have been designed in a $0.8\text{-}\mu m$ CMOS technology. The datapath and the synchronizing networks are marked in this figure in order to illustrate their parts of the silicon area.

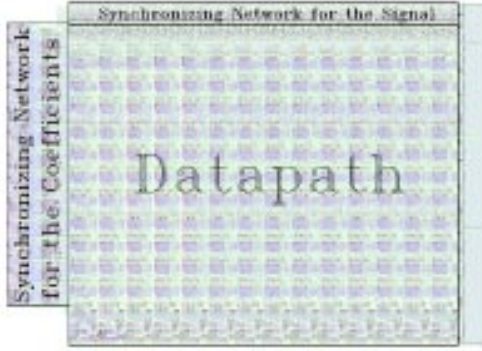Fig. 3: Layout of an 3taps FIR-Filter
($w_c = w_x = 8bit$, $N = 2$)

The use of a signed-digit number representation allows to replace the full-adder cells needed due to the sign extension in two's complement arithmetic by half-adder cells. As the fourth design criterion has been followed, this replacement does not cause any additional routing overhead and thus results in a further reduction of silicon area of about 20%. An architecture based on this number representation will be presented in some more detail for the following example.

3.2. Matched Filters for Spread Spectrum Receivers

A bank of matched filters (Hybrid Architecture) has been shown to be a cost efficient solution [7] for rapid code acquisition in spread spectrum receivers. The optimum degree of parallelism for those architectures varies as well as the specified chip-rate according to the application [2, 7]. However, in contrast to the application described above the number of filter taps as well as the time-sharing factors N that are of interest are typically much larger. In addition to that the wordlength of the coefficients w_c is 1bit.

The serial-in/parallel-out SFG in Fig. 4a can be mapped to an hSFG as indicated in Fig. 4b in order to derive a class of appropriate filters. The shown design-flow is related to the realistic example of a filter of order $M = 128taps$, and the input signal wordlength $w_x = 6bit$ for a time-sharing factor of $N = 16$.

Assuming a signed-digit number representation the required generalized full-adders and half-adders can be realized by compact inverting full-adders and half-adders. Accordingly every second partial product has to be inverted. As the coefficients determine the signs of the partial products, they have to be fed to the input terminals of the least significant full-adders of each adder

stage. Fig. 4c shows the resulting hSFG after cut-set retiming including internal wordlengths. Therein some inversions have been moved together with some pipeline latches into the synchronizing networks.
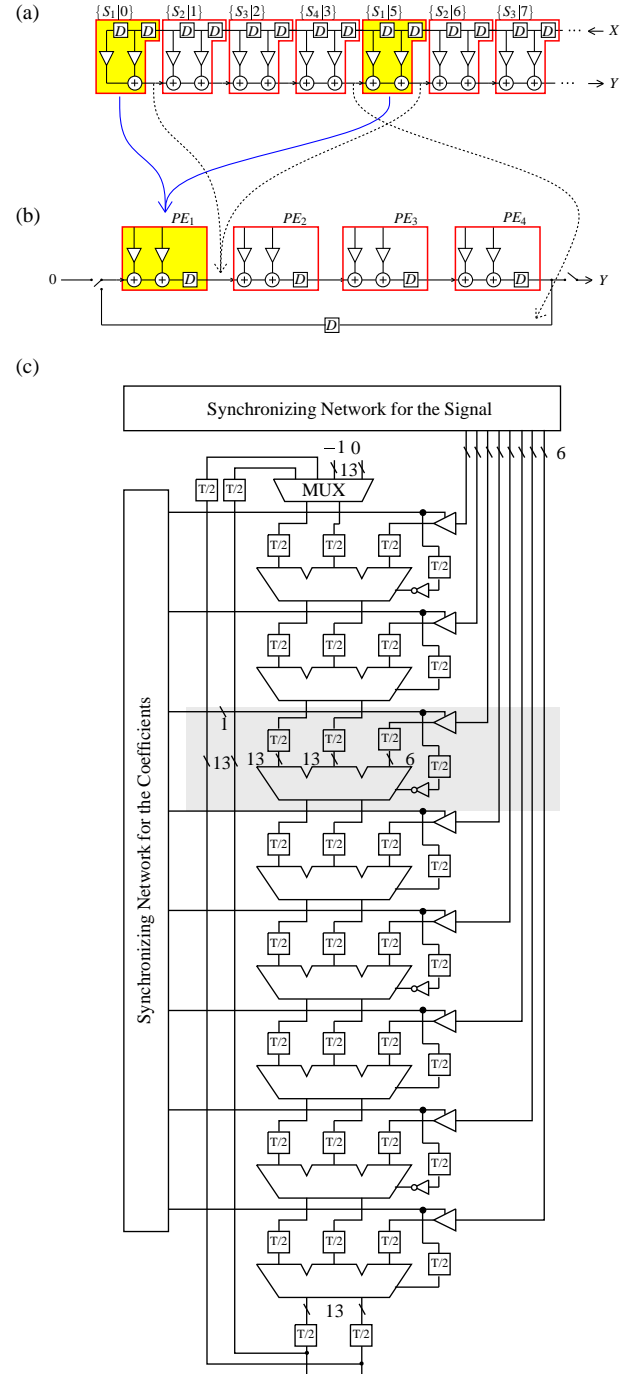

Fig. 4: (a) aSFG (b) hSFG (c) Architecture of the Datapath for $M = 128taps$, $w_x = 6bit$, $w_c = 1bit$, and $N = 16$

Fig. 5 shows the slice highlighted in Fig. 4c on the gate level containing three different basic cells (dashed boxes). The shown modification of the half-adder basic cell for the most significant
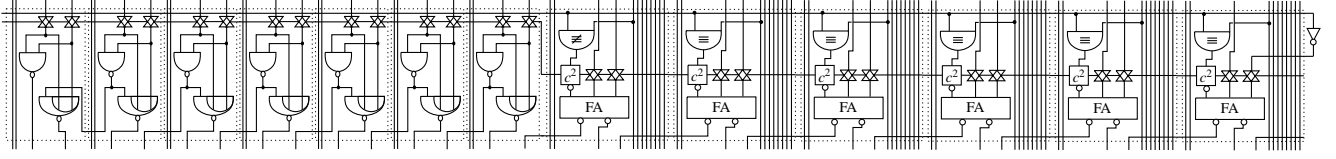
Fig. 5: Schematic of one Stage of the Datapath for Rapid-Code Synchronization in Spread Spectrum Receivers

weight saves one half-adder basic cell. In contrast to actual approaches [8] no further post-addition of a negative constant is necessary.

After any N time cycles the two vectors with opposite signs have to be combined at the output by a vector merging adder.

The synchronizing network for the input signal can be realized by an array of dynamic storage elements controlled by a pointer signal passing through a cyclic shift register. The synchronizing network for the coefficients can be realized by an array of storage elements with one part for buffering the new value and another part for keeping the actual value.

Fig. 6 shows the Layout of the datapath for the example in Fig. 4 occupying a silicon area of $0.13mm^2$ in a 0.8-μm CMOS technology.
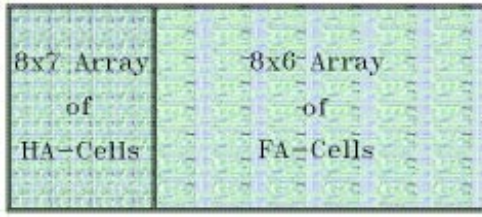

Fig. 6: Layout of the Datapath for the Matched Filter

In Hybrid Architectures only one synchronizing network for the signal has to be implemented for any number of parallel matched filters leading to significant savings in silicon area. Based on the described architecture code acquisition units for several numbers N_p of parallel filters, several time sharing factors N, and a spreading factor of 1024 have been automatically tiled with only a small set of basic cells.
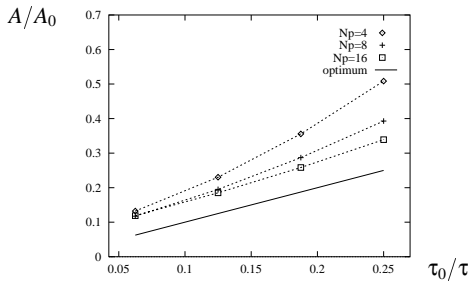

Fig. 7: Silicon Area vs. Throughput Rate

Fig. 7 shows the relative silicon area A/A_0 vs. the relative throughput rate τ_0/τ of the designed architectures. A_0 and τ_0 are the corresponding values of realizations without time sharing.

4. CONCLUSION

A systematic mapping approach leading to efficient VLSI-architectures that matches given system specifications by means of time-sharing has been presented. This approach has been applied to derive two novel classes of FIR-filter architectures for different applications. These architectures enable to trade hardware for speed efficiently with only a small number of basic cells and therefore the design effort is kept small. The evaluation of implemented code acquisition units for direct sequence spread spectrum receivers shows the high gain in silicon area reachable by the use of the proposed method.

Current investigations deal with applying the design approach to multirate systems for subband coding, for example, as well as two dimensional FIR-filters.

REFERENCES

[1] Z.-J. Mou and P. Duhamel "Short-Length FIR Filters and Their Use in Fast Nonrecursive Filtering," IEEE Transactions on Signal Processing, vol. 39, no. 6, pp. 1322-1332, June 1991

[2] C. Chien and R. Jain and E. G. Cohen and H. Samueli "A Single-Chip 12.7 Mchips/s Digital IF BPSK Direct Sequence Spread-Spectrum Transceiver in 1.2 μm CMOS," IEEE Journal of Solid-State Circuits, vol. 29, no. 12, pp. 1614-1623, December 1994

[3] Liang-Gee Chen, Yeu-Shen Jehng and Tzi-Dar Chiueh "Pipeline Interleaving Design for FIR, IIR, and FFT Array Processors," Journal of VLSI Signal Processing, vol. 10, pp. 275-293, October 1995

[4] S. Y. Kung "VLSI ARRAY PROCESSORS," Englewood Cliffs, NJ: Prentice–Hall, 1988

[5] K. K. Parhi and C.-Y. Wang "Synthesis of Control Circuits in Folded Pipelined DSP Architectures," IEEE Journal of Solid-State Circuits, vol. 27, no. 1, pp. 29-43, January 1992

[6] T. G. Noll "Carry–Save Architectures for High-Speed Digital Signal Processing," Journal of VLSI Signal Processing, vol. 3, pp. 121-140, 1991

[7] E. A. Sourour and S. C. Gupta "Direct-Sequence Spread-Spectrum Parallel Acquisition in a Fading Mobil Channel," IEEE Transactions on Communications, vol. 38, no. 7, July 1990

[8] Hwan-Rei Lee, Chein-Wei Jen, and Chi-Min Liu "A New Hardware-Efficient Architecture for Programmebale FIR Filters," IEEE Transactions on Circuits and Systems-II, vol.43, no. 9, pp. 637-644, September 1996